AD-A057 312 ROCHESTER UNIV NY DEPT OF COMPUTER SCIENCE F/G 12/1
FAST DISPLAY OF WELL-TESSELATED POLYHEDRA, (U)
MAR 78 C M BROWN N00014-75-C-1091
UNCLASSIFIED TR-23 NL

1 OF 1
ADA
057312

電腦科學

LEVEL II

rochester

Department of Computer Science
University of Rochester
Rochester, New York 14627

78 07 17 140

(11)

(6)

Fast Display of Well-Tesselated Polyhedra

(10) Christopher M. Brown
Computer Science Department
The University of Rochester
Rochester, NY 14627

(14) TR23

(11) March 1978

(12) 23p.

This research was supported in part by the Alfred P. Sloan Foundation under Grant No. 74-12-5, and in part by the Defense Advanced Research Projects Agency, monitored by the ONR under Contract No. N00014-75-C-1091.

(15)

Abstract

Well-tesselated polyhedra are a subclass of those polyhedra whose faces are triangular, possibly transparent, and all visible from a single origin point. They admit of simple hidden-line and -surface algorithms. In a raster graphics environment, the algorithms yield a priority ordering for painting entire faces. This order is invariant over radial translations of polyhedral vertices (therefore invariance extends to the perspective transformation).

Specific methods for creating well-tesselated polyhedra are given, and general constraints defining them are stated. An efficient hidden surface algorithm is presented; a simpler method for the case of opaque faces also produces hidden line drawings. Demonstrations of correctness are provided.

Keywords and Phrases: three-dimensional computer graphics, raster graphics, hidden surface elimination, hidden line elimination, polyhedral objects, geodesic constructions.

CR Categories: 3.10, 3.41, 8.2.

1. Overview

A class of well-tesselated polyhedra may be defined which is large enough to be useful for many computer graphics applications and for which very simple hidden-line and -surface algorithms exist. These generally nonconvex polyhedra have triangular, possibly transparent faces, each of which is visible from some origin point (such polyhedra are called museum-viewable). The faces of well-tesselated polyhedra also meet a local geometric criterion defined in Section 2. A polyhedron which is, well-tesselated remains so if its vertices are translated radially from its origin point, as its faces stretch and tilt to accommodate. Some geodesic dome [5] constructions yield well-tesselated polyhedra (see Section 4.), examples of which appear in Figure 1.

<<INSERT FIGURE 1 HERE>>

Figure 1. A sampling of well-tesselated polyhedra, shaded by a basic technique.

Sutherland, Sproull, and Schumacker [7] have shown that hidden surface (line) algorithms have in common the operation of sorting. The two algorithms given here (one a special case of the other) illustrate this observation particularly well; to display N faces the simpler algorithm just sorts N numbers. Being so simple, the algorithms themselves are of limited novelty; they are quite similar to subcases of the algorithm of Newell, Newell, and Sancha (NNS) [6]. What is more interesting is that such simple treatment suffices for a useful class of objects.

The algorithms here (like that of NNS) are list-priority algorithms; they compute their results in object space to all available accuracy, and are limited in output accuracy only by the display medium. Like NNS, they depend on a raster graphics output device, in which an image memory raster is repetitively scanned out onto a viewing screen. Image information may be painted into the raster, thereby overwriting or combining with the previous contents; this capability allows for simulation of obscuration and transparency.

The algorithms of this paper differ from that of NNS in three related respects.

First, not only will the algorithms deal correctly with well-tesselated polyhedra, but the painting priority order they derive is invariant over the radial deformation transformations of moving polyhedral vertices in or out radially from the origin as the faces stretch and tilt to accommodate. Thus, a single application of an algorithm yields a painting order correct for an infinite set of related polyhedra; special cases of related polyhedra include those resulting from pure perspective distortion and those that simulate rigid rotation to some accuracy (see Section 3.)

Secondly, there is an underlying semantic difference. To get a tentative order for painting faces onto an image raster, NNS sort the faces on the maximum distance (depth) they attain from the viewpoint; nearer faces are to overpaint farther ones. The algorithms here sort faces instead on the maximum angle (at the origin) they attain with the viewing direction. This explains why the priority order is invariant over radial deformations, which do not change these angles.

Thirdly, both the algorithms given here are much simpler than that of NNS. The initial sort on depth is inadequate for general objects, which may require faces to be divided and further tests (as many as six in NNS) to be applied before a correct priority order is obtained. In contrast, the opaque face algorithm presented here performs only the initial sort and needs no fixups; with transparent faces, ties in the initial sorted order must be resolved by one test. Face-dividing is never necessary.

The algorithms allow production of shaded surface visualizations if faces are painted according to some illumination and reflection model; faces of varying degrees of transparency may be defined. A bonus is that since faces are never divided, a hidden-line drawing may be produced by painting edges of faces black and their interiors white.

In Section 2 well-tesselated polyhedra are defined and discussed. In Section 3 the algorithms are described informally and their timing is considered. In Section 4 constructions for well-tesselated ancestors are presented. The appendix has a more careful statement of tests and algorithms, and demonstrations of correctness.

2. Well - Tesselated Polyhedra

Well-tesselated polyhedra arise from polyhedra inscribed in the sphere, which themselves come from certain tesselations of the unit sphere centered at the origin. The tesselations of interest are triangulated graphs embedded in the sphere. They produce a set of spherical triangles or patches partitioning the surface of the sphere. Tesselations induce (spherically) inscribed polyhedra in an obvious way: the three vertices of each patch determine a plane face of the inscribed polyhedron. The inscribed polehedron is well-tesselated if its tesselation meets the Angle Condition (AC): around any tesselation vertex on the sphere, the angle between two nonadjacent patch edges is not less than pi/2, and the angle between any adjacent patch edges is not greater than pi/2. Finally, well-tesselated polyhedra may be derived from other well-tesselated (possibly inscribed) polyhedra by translating vertices radially with respect to the sphere center, or origin; incident edges and faces are carried along. Thus the tesselation defines a set of half-lines which radiate from the origin through each patch vertex; the polyhedral vertices may lie anywhere on these half-lines.

Strong geometric coherence [7] in well-tesselated polyhedra allows simple hidden-line and -surface algorithms. A tesselation induces a set of infinite pyramids throughout 3-space (by radial projection of its patches) which are partially ordered in depth. The correctness of the algorithms depends heavily on the fact that polyhedral faces are connected to their neighbors; along with constraints on pyramids enforced by the AC, this guarantees that the pyramidal partial ordering will remain usable for polyhedral faces as well. The function of the AC is to prevent certain overlaps between faces which render the simple algorithms inadequate. The effects of the locally-defined AC on the general appearance of tesselations is hard to characterize, but some facts are known. The degree of a tesselation vertex meeting the AC must be between 4 and 8; patch sizes may vary, but only gradually, over the sphere; patches with excessively small or large angles are unlikely. If a polyhedron is not well tesselated, the algorithms may work anyway; however, violations of the AC allow the construction method above to produce polyhedra for which the algorithms fail (see Section 3.) The interested reader is referred to the appendix for a more careful look at tesselation conditions and their relation to the algorithms.

3. Hidden - Surface Algorithms

The imaging model is that the viewpoint is on the positive z axis of a right-handed Cartesian coordinate system. The image plane is the z=0 plane, with x increasing to the right and y increasing upwards as seen from the viewpoint. The polyhedra are centered on the origin; their images are their projections onto the image plane through the viewpoint. The hidden surface algorithms determine a priority order for painting faces into an image raster representing the image plane. The opaque face algorithm is a subcase of the general algorithm allowing for transparent faces; more detailed and precise statements of both appear in the appendix.

The opaque face algorithm. Discard back faces (those whose surface normals point away from the viewpoint by more than pi/2 radians). Obtain the painting priority order for the faces by sorting them in order of decreasing maximum angle at the origin between any face point and the viewing direction (the z axis). (If the sphere is replaced by a globe, and if the viewpoint is imagined to be above the north pole, then the sort is on the minimum latitude of the spherical projection of a face.) Ties may be broken at random.

The general algorithm. Here, back faces may no longer be discarded. The same decreasing maximum angle sort is performed on faces, but now ties may not be broken at random. If two faces share an edge and are tied for minimum z, then paint the one first which is farthest from the viewpoint in the sense of being on the far side of the plane formed by their common edge and the origin.

If it is performed on the inscribed version of the polyhedron, the sorting operation of these algorithms becomes the initial sort of the NNS algorithm, since the point of maximum face depth then corresponds to the point of maximum angle. This similarity is more accidental than basic; generally the painting order for polyhedral faces yielded by NNS and by these algorithms will differ.

Figure 2 presents cases in which the algorithms fail; the image plane is the plane of the page, the viewing direction is normal to the page. The polyhedron shown in Figure 2A is not well-tesselated.. It arises from a tesselation based on lines of longitude and the equator of a globe, if the viewpoint is in the equatorial plane; the AC is violated at the poles. Figure 2B shows a polyhedron obtained by translating the vertex X radially (approximately toward the viewer's left eye); face P now occludes face V. However, P and V are tied at the north pole for maximum angle with the viewing direction. Since ties are broken at random in the opaque-face algorithm, it may fail, incorrectly painting V after (over) P. Another case is illustrated in Figure 2C. Only two patches on the sphere are shown, but it is fairly clear that the AC must be violated somewhere if a tesselation includes them. On the sphere, lines of constant angle with the viewing direction are circles concentric with the sphere boundary; it is seen that patch (face) V has a smaller maximum angle, and will thus be painted later by either algorithm. Figure 2D shows particular faces arising from the patches; P has been projected farther than V, and is closer to the viewpoint everywhere. The later (over) painting of V is a mistake.

<<INSERT FIGURE 2 HERE>>

Figure 2. The inscribed polyhedron in (A) is not well-tesselated; (B) shows a potential face misordering. The patches in (C) also cause a potential misordering (D).

Polyhedra have a well-defined notion of inside and outside, and for museum-viewable polyhedra it is easy to tell whether a face presents its inner or outer side to the viewpoint. Shading algorithms can thus easily cater to the inner/outer difference: steps may be taken to treat the interior and exterior alike, faces may have different reflectivities, transparencies, or colors on different sides, etc. Painting opaque faces with black edges and white interiors yields a hidden-line drawing.

The time complexity of the algorithms is theoretically dominated for large N (number of faces) by the sort, which is taken to be O(NlogN.) The difficulty of the geometric calculations depends on representations used. Removal of back faces involves at worst examining each face once and doing the geometry (equivalent to a few vector subtractions and dot products, and a cross-product) to form the plane equation and test on which side the viewpoint lies. Finding the minimum z involves at most normalizing the vertex vectors of each face and

making two comparisons. Breaking ties in the general algorithm involves examining each face at most once, and occasionally locating a neighbor and performing operations similar to back face removal calculations. In practice, the running time has been small relative to geometric calculations. On a minicomputer, using a high-level language with microcoded floating point, the shaded, ordered faces for each polyhedron of Figure 1 (N is about 160) were produced in about four seconds, the hidden-surface algorithm taking about one second, the shading calculation about three seconds. The operations of creating polyhedra, rotating them, and sending the faces to the display processor all take up to an order of magnitude longer.

It is worth noting, however, that for some polyhedra with a large number of faces, it could be worthwhile to avoid recomputing the priority order between rotations. This can be done by considering the painted polyhedron to be an approximation to an ideal object which it encloses. The enclosing polyhedron has rubber faces, and its vertices can move in and out along fixed radial half-lines. As the object of interest rotates inside the non-rotating but deformable enclosing polyhedron, it distorts the rubber faces into new (rotated) approximations via radial deformation. The original painting order is correct for all these deformations. The usefulness of this technique depends on the characteristics of the object of interest and the tesselation; the constructions of Section 4 have good properties for this application.

For the algorithms to perform correctly, well-tesselated polyhedra are sufficient but not necessary. The algorithms can be expected to perform tolerably well for polyhedra arising from other constructions (such as that of Fuchs et al. [4]) if the polyhedra are museum-viewable and sufficiently smooth.

4. Two Constructions for Well - Tesselated Polyhedra

Certain geodesic constructions [5] give well-tesselated inscribed polyhedra; they are discussed in more detail in [1]. The idea is to inscribe an icosahedron in the sphere, then to subdivide each icosahedral face into triangular subfaces whose vertices are projected onto the sphere. The icosahedron has 12 vertices, 20 faces, and 30 edges. If each edge is divided into n new edges, the polyhedra resulting from the geodesic constructions below have 10*n**2 + 2 vertices, 20*n**2 faces, and 30*n**2 edges.

Clinton [3] gives seven methods for subdividing the initial icosahedral faces. The simplest of these is to divide each one into n**2 congruent equilateral triangles by dividing each edge into n equal lengths and connecting the division points by lines parallel to the edges of the face. During projection onto the sphere, the central subfaces of an icosahedral face are projected farther than those near an icosahedral vertex, and so produce larger inscribed faces; this symptom is more acute with finer subdivisions.

To produce more nearly congruent inscribed faces, the icosahedral edge subdivisions may be made to subtend equal angles on the sphere. After the division points on the edges have been thus located, they are connected by lines parallel to the edges of the icosahedral face. Since the division points are not equally spaced along the edges, the lines through them do not meet in points, but in small triangles. The centroid of each small triangle is projected to become a vertex. This method yields a substantial improvement in face uniformity. For four divisions per icosahedral edge, the ratio of largest to smallest solid angle subtended by inscribed faces is 1.517 for the first method, 1.146 for the second.

The following results are useful in icosahedron construction. Define:

t = the golden ratio = (1 + sqrt(5))/2
a = sqrt(t) /(5**(1/4))
b = 1/(sqrt(t) * (5**(1/4))).

The angle subtended by an icosahedral edge is arccos(sqrt(5)/5) radians. The twelve vertices may be placed (in Cartesian co-ordinates) at

$(0, \pm a, \pm b)$
$(\pm b, 0, \pm a)$
$(\pm a, \pm b, 0)$.

5. Conclusions

A class of triangular-faced, museum-viewable polyhedra may be defined by radial vertex translation acting on spherically-inscribed polyhedra which themselves arise from tesselations of the sphere meeting a local Angle Condition. This class of well-tesselated polyhedra, while constrained in its construction, is useful for representing certain well-behaved physical objects, and has found application as a visualization of 3-D histograms [2]. The hidden line and surface problem is easy for well-tesselated polyhedra, since the constraints eliminate much of the computation necessary in general object-space or list-priority hidden surface algorithms.

A simple algorithm exists which establishes a priority painting order for well-tesselated polyhedra with transparent faces; a simpler method exists for the case of opaque faces (which gives hidden-line drawings as a special case). An unsophisticated minicomputer implementation of the opaque face algorithm takes a few seconds when applied to polyhedra with a few hundred faces. The priority order for painting is invariant over transformations of the polyhedra involving radial translation of vertices; this includes the perspective transform and in some cases an approximation to rotation. Thus running the algorithm once gives the priority order for an infinite number of related polyhedra for an infinite number of viewing distances.

Acknowledgements. Many colleagues have assisted these efforts. Deserving special mention are K. Sloan, who first stated the AC and whose constructive criticism has been

invaluable, P. Selfridge for graphics production, A. Requicha, J. Nabielsky and R. Sproull for critical reading.

References

1. Brown, C.M. Representing the orientation of dendritic fields with geodesic tesselations. TR13, Department of Computer Science, University of Rochester, Rochester N.Y., 1977.

2. Brown, C.M. Two descriptions and a two-sample test for 3-D vector data. Submitted for publication, 1978.

3. Clinton, J.D. Advanced structural geometry studies part I: polyhedral subdivision concepts for structural applications. NASA CR-1734/35, September 1971.

4. Fuchs, H., Kedem, Z.M., and Uselton, S.P. Optimal surface reconstruction from planar contours. Comm. ACM 20, No. 10, October 1977, 703-712.

5. Fuller, R.B. Building construction. U.S. Patent No. 2,682,235, 1951.

6. Newell, M.E., Newell, R.G., and Sancha, T.L. A new approach to the shaded picture problem. Proceedings of ACM National Conference, 1972.

7. Sutherland, I.E., Sproull, R.F., and Schumacker, R.A. A Characterization of ten hidden-surface algorithms. Computing Surveys 6, No. 1, March 1974, 1-55.

Appendix

This appendix is relatively self-contained, but for brevity it relies on the main body of the paper for some basic terms and concepts. It contains explicit geometrical tests alluded to in the text, Angle and Edge Conditions for tesselations, detailed statement of the algorithms, and proofs that the algorithms are correct. It relies heavily on the technique of reasoning about sets of polyhedral faces by means of reasoning about the infinite pyramids (radial projections of patches) which contain them, and about pyramids by means of the patches which produce them. This gives rise to locutions such as "painting a patch," meaning "painting any face whose spherical projection is a patch." A face is the face of a particular polyhedron. A patch implicitly references infinitely many faces. To avoid confusion, the vertex of a patch is hereafter called a pvertex and the edge of a patch a pedge: edge and vertex refer to polyhedral edges and vertices. Half-lines from the origin through pvertices are called vertex directions, since vertices must lie in them. Each patch or face P has three edge-neighbors, or e-nbrs (which share a (p)edge with P) and a number of vertex-neighbors, or v-nbrs (which share exactly one (p)vertex with P.) A non-neighbor, or non-nbr, shares no (p)vertices.

1. Coordinates

Let a spherical coordinate system (rho, theta, phi) have a common origin with the Cartesian coordinates. Phi is the polar angle, varying from 0 at the positive z axis to pi at the negative z axis. Points in the image plane have phi = pi/2. The angle theta varies from 0 to 2*pi counterclockwise from the positive x axis as seen from the viewpoint. If a globe is placed with its center at the origin, with the z axis piercing the North and South poles, then the image plane cuts along the equator, cones of constant phi cut lines of latitude, and planes of constant theta cut lines of longitude.

Points in space are Cartesian 3-vectors x=(x,y,z), or 3-vectors p=(rho,theta,phi). pvertices are spherical 3-vectors (1,theta,phi). The functions theta(.) and phi(.) give the theta and phi coordinates of a point. Then for a point,

```
(x,y,z) = (rho*sin(phi)*cos(theta), rho*sin(phi)*sin(theta),
           -rho*cos(theta))
rho = sqrt(x**2 + y**2 + z**2)
theta = arctan(y/x)
phi = arccos(z/rho).
```

A line of sight is a half-line emanating from the viewpoint passing through the image plane. The z axis and any other line of sight determine a plane which cuts the sphere in a line of constant theta, and thus is perpendicular to lines of constant phi. This fact is important in the sequel.

2. Well - Tesselated Polyhedra

Definition: Well - Tesselated Polyhedron
A polyhedron is well-tesselated iff its tesselation meets the Angle Condition.

Definition: The Angle Condition (AC) for Tesselations
Around any pvertex, the angle between any two nonadjacent pedges is not less than pi/2, and the angle between any adjacent pedges is not greater than pi/2.

Theorem I:
If the AC is met, then if d is the degree of the tesselation graph (maximum number of patches around a pvertex), 4=<d=<8.
Proof:
Start at any pedge and label the angles subtended by patches counterclockwise as theta(0), theta(1), etc. to theta(d-1).
Then AC implies
theta(i) + theta((i+1) mod d) >= pi/2;
Hence,
2*Sum of theta over i>= d*pi/2.
But
Sum of theta over i must be equal to 2*pi,
so 2*pi >= d*pi/4, and d =< 8.
Similarly, AC implies theta(i) =< pi/2, so
Sum of theta over i =< d*pi/2,
so 2*pi =< d*pi/2, or d => 4.□

The Edge Condition is in terms of face edge lengths, not angles; it gives a different insight into the constraints on well-tesselated polyhedra.

Definition: The Edge Condition (EC) for Tesselations
All pedges are shorter than one radian, and the ratio of longest to shortest pedges of any patch is less than 1.27.

Theorem II:
The Edge Condition implies the Angle Condition.
Proof:

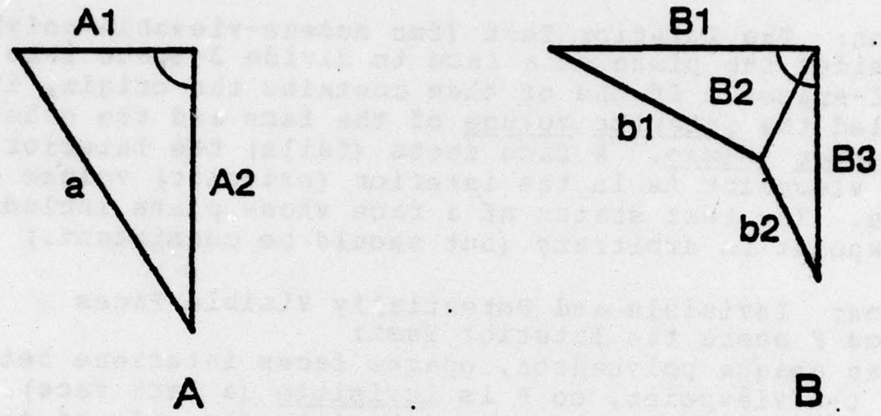


Figure 3. Spherical triangles.

The AC requires that two adjacent pedges subtend at most pi/2. Figure 3A shows this case. A symmetry argument shows that the minimum ratio of sides allowing the maximum subtended angle is obtained when A1 = A2 = A. Then by spherical trigonometry,

a = arccos(cos(A)**2).

The ratio a/A monotonically decreases with increasing A, from a maximum (in the limit of plane triangles) of sqrt(2). At A = 1 radian the ratio is 1.2745; thus a side ratio greater than 1.27 is needed to violate the adjacent-pedge AC for pedges shorter than one radian, but is forbidden by the EC.

The AC also requires two nonadjacent pedges to subtend angle at least pi/2. Figure 3B shows this case, which is handled similarly. Here it is found that B1 = B2 = B3 = B, b1 = b2 = b, and the minimum ratio forcing this minimum subtended angle is

b = arccos(cos(B)**2 + 2*cos(pi/4)*(sin(B)**2).

The ratio B/b monotonically increases with increasing B, from a minimum (in the limit of plane triangles) of 1/sqrt(2-sqrt(2)) = 1.3066. Thus the EC forces the non-adjacent pedge AC. □

The minimum ratio of the EC, 1.27, forces the AC if all pedges are smaller than one radian. A calculation shows that the ratio 1.3066 forces the AC if all pedges are less than .5791 radians.

3. Algorithms and Tests

Algorithm CreatePolyhedronFromTesselation

1. Distinguish a single point in every vertex direction; the point is a vertex.
2. For every patch, construct a plane polyhedral face between the three vertices in the vertex directions

which contain the pvertices of the patch.

Definition: The Interior Test (for museum-viewable polyhedra)
Consider the plane of a face to divide 3-Space into two open half-spaces. If one of them contains the origin, it is called the interior volume of the face and the other is the exterior volume. A face meets (fails) the Interior Test iff the viewpoint is in the interior (exterior) volume of a face. The test status of a face whose plane includes the viewpoint is arbitrary (but should be consistent.)

Definition: Invisible and Potentially Visible Faces
Iff a face F meets the Interior Test:
In an opaque polyhedron, opaque faces intervene between F and the viewpoint, so F is invisible (a back face).
If transparent faces exist, the interior side of the face is potentially visible (i.e. visible except for obscurations.)
Iff a face fails the Interior Test:
In an opaque polyhedron, F is potentially visible.
If transparent faces exist, the exterior side of F is potentially visible.

The function maxphi(.) gives the maximum phi value attained by a face or patch; only the (p)vertices need be tested to determine it.

Algorithm PaintOpaquePolyhedron

Input: The (opaque) face set OpaqueFaces of a well-tesselated polyhedron.

1. Discard a face F from OpaqueFaces if F meets the Interior Test.
2. Associate maxphi(F) with each F remaining in OpaqueFaces.
3. Sort OpaqueFaces into descending order of associated maxphi(F), breaking ties at random.
4. Paint the members of OpaqueFaces in sorted order.

Algorithm PaintGeneralPolyhedron

Input: The set PolyhedronFaces of (possibly transparent) faces of a well-tesselated polyhedron.

1. For each face F in PolyhedronFaces, associate maxphi(F) with F.
2. Form an ordered list L of sets of faces. Each set in L has as members all faces with a particular value of maximum phi, and the list is in order of descending common maximum phi value.
3. In order, submit the sets in L to Subroutine PaintAPhi.

PaintAPhi uses the Priority Test defined below to break ties in the initial sort. Stated for patches, it holds for faces. A posterior e-nbr should always be painted before its prior e-nbr, since the prior patch (hence its entire pyramid) is between the viewpoint and the posterior patch (pyramid) for every line of sight passing through both. This fact is useful in the proof of Lemma VI.

Definition: The Priority Test

The origin and the two shared pvertices of e-nbrs determine a plane dividing 3-Space into two open half-spaces. If the viewpoint is (is not) in the same half-space as one of the patches, that patch is the <u>prior</u> (<u>posterior</u>) patch. If the viewpoint is in the plane, the test gives no information.

<u>Subroutine PaintAPhi</u>

Input: a set IdentPhi of faces with identical maximum phi.

1. Associate a classification with each face, initially "unclassified."
2. Repeat this step until no unclassified faces remain: let F be an unclassified face.
   a. If F has no e-nbrs, classify it "isolated"
   b. else if F has an e-nbr classified "prior" ("posterior"), similarly classify the other e-nbr (if any); classify F the opposite, viz. "posterior" ("prior")
   c. else if the Priority Test can determine that F is prior (posterior) for one of its e-nbrs, so classify F and classify its e-nbr(s) the opposite, viz. "posterior" ("prior")
   d. else classify F "isolated."
3. Paint faces:
   a. Paint posterior faces in any order.
   b. Paint prior faces in any order.
   c. Paint isolated faces in any order (this step can be done any time relative to steps 3a and 3b).

<u>4. Correctness of the Algorithms</u>

Both algorithms paint all potentially visible faces. The proofs involve demonstrating that no face is wrongly overpainted by another face. The strategy will be to classify a patch V as <u>suspicious</u> with respect to an already painted patch P if (roughly) V overlaps P, is partly behind P, and has no point of greater phi than P. These criteria have the flavor of traditional hidden-surface tests on faces. If V is not suspicious it cannot possibly affect P when V is painted by either algorithm. A suspicious patch V is <u>dangerous</u> to P if it (i.e. any of its faces) can actually wrongly overpaint P (i.e. any of P's faces) in the course of a particular algorithm. A patch that is dangerous in the algorithm PaintGeneralPolyhedron is <u>PolyDangerous</u>; in the algorithm PaintOpaquePolyhedron it is

<u>OpaqueDangerous</u>. By definition, if there are no dangerous patches for an algorithm, the algorithm is correct. It will be shown that the AC prevents many suspicious patches from existing, and that those that remain are not PolyDangerous or OpaqueDangerous.

Definition: Behind

A patch V is <u>behind</u> P (in an overlap interval) if: there exists a closed overlap interval [theta 0, theta1], theta0<theta1, such that phi(v)>=phi(p) for all points p and v with p in P, v in V, and theta0 =< theta(p) = theta(v) =< theta1. If V is not behind P in an overlap interval, then P is behind V (or V is in front of P).

If patch V is behind patch P, then along any line of sight in the overlap, the pyramid of patch P is between that of patch V and the viewpoint. Thus if a face of one of these patches obscures a face of the other, it must be the face of P which does the obscuring, and hence which should be painted later than the patch of V.

Definition: Suspicious

A patch V is suspicious with respect to a patch P if both:

1. maxphi(V) =< maxphi(P)
2. V is behind P in some overlap interval.

Figure 2C shows a particular case of a patch V which is both suspicious and dangerous with respect to P.

Theorem III:
If a patch V is not suspicious with respect to a patch P, it is not dangerous to P.
Proof:
If maxphi(V) > maxphi(P), both algorithms paint V before P, so V cannot overpaint P. If there is no overlap in theta, V cannot overpaint P by Fact I. If there is overlap but V is not behind P in the overlap, then P is behind V. Thus if V overpaints P it does so correctly, and so V is not dangerous to P.□

The results below are independent of the position of the viewpoint on the z-axis. Theorem IV explicitly shows that perspective distortion does not necessitate a resorting of the faces.

Theorem IV.
Perspective distortion leaves vertex directions invariant.
Proof:
The general perspective distortion operating on a point $\underline{x}$ may be expressed for some focal length constants f,g, and h as

$$\underline{x}' = (1/(1 + z/f + y/g + x/h))\underline{x}.$$

Thus the magnitude of x is changed by perspective, but not its direction.□

4.1. Correctness of PaintOpaquePolyhedron

Lemma I:
An e-nbr E of a patch P is not OpaqueDangerous to P.
Proof:
If E is invisible it is a back face not painted by PaintOpaquePolyhedron, hence it cannot be dangerous. If E is potentially visible, the edge common to E and P must separate the images of P and E on the image plane, so E cannot overpaint P.□

Lemma II:
In a well-tesselated polyhedron, no v-nbr V of a patch P is suspicious with respect to P.
Proof:

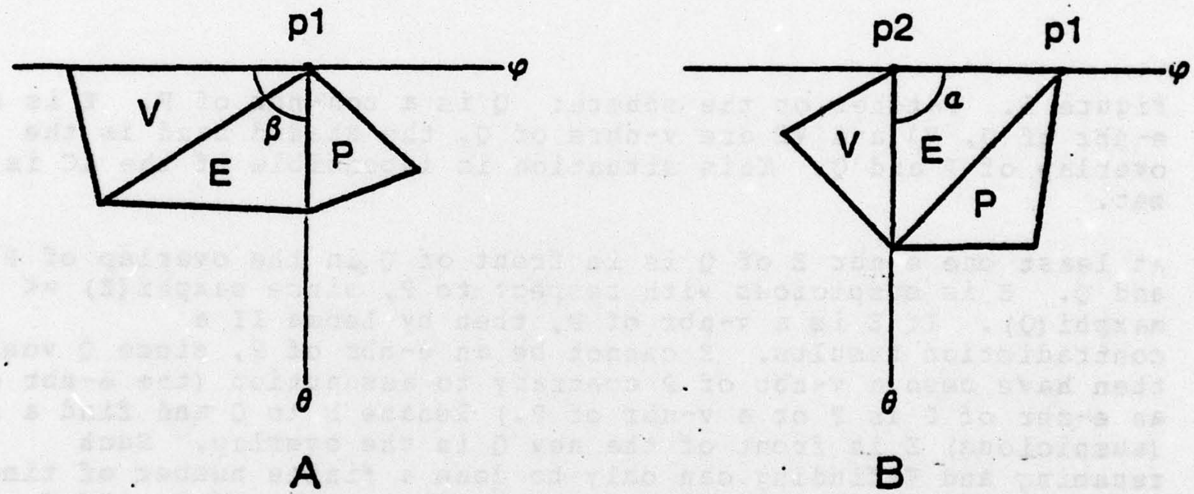


Figure 4. Patches on the sphere: theta is a line of sight (constant theta), phi a line of constant phi. V is a v-nbr of P, E is an e-nbr of P.

Case I: the pvertex p1 of maximum phi is the same for both P and V. In Figure 4A, the AC guarantees the angle beta >= pi/2, so any overlap can be at most the singleton set theta.

Case II: the pvertex p1 of maximum phi for P is different from p2, the pvertex of maximum phi for V. Alpha, the angle subtended by E in Figure 4B, must be <= pi/2 by the AC, so any overlap can be at most the singleton set theta.□

Lemma III:
In a well-tesselated polyhedron, no non-nbr Q of a patch P is suspicious with respect to P.
Proof:
Suppose Q, a non-nbr of P, were suspicious with respect to P

(Figure 5).

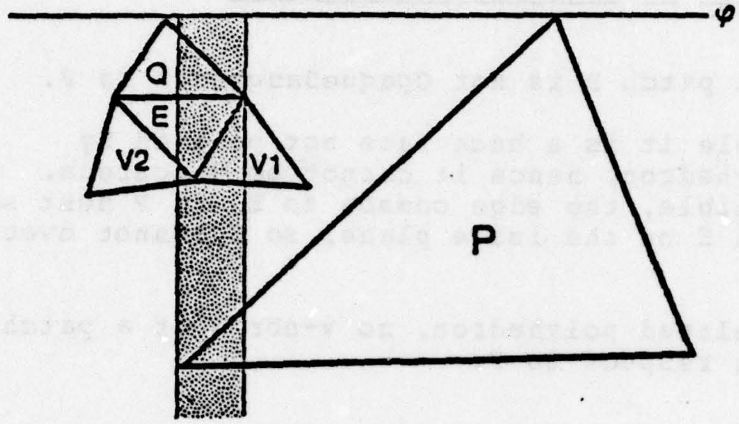


Figure 5. Patches on the sphere: Q is a non-nbr of P. E is an e-nbr of Q, V1 and V2 are v-nbrs of Q, the shaded band is the overlap of P and Q. This situation is impossible if the AC is met.

At least one e-nbr E of Q is in front of Q in the overlap of P and Q. E is suspicious with respect to P, since maxphi(E) =< maxphi(Q). If E is a v-nbr of P, then by Lemma II a contradiction results. E cannot be an e-nbr of P, since Q would then have been a v-nbr of P contrary to assumption (the e-nbr of an e-nbr of P is P or a v-nbr of P.) Rename E to Q and find a new (suspicious) E in front of the new Q in the overlap. Such renaming and E-finding can only be done a finite number of times before the current E becomes an e-nbr or a v-nbr of P. If E becomes an e-nbr, then the current Q is a suspicious v-nbr, a contradiction by Lemma II. If it becomes a v-nbr, a similar contradiction arises, since E is suspicious itself.□

Theorem V:
PaintOpaquePolyhedron is correct; in a well-tesselated polyhedron, no patch Q is OpaqueDangerous with respect to a patch P.
Proof:
If Q is an e-nbr of P, it is not OpaqueDangerous to P by Lemma I and Theorem III. If Q is a v-nbr or non-nbr of P, then by Lemmas II and III it is not even suspicious; by Theorem III it cannot be OpaqueDangerous to P.□

4.2. Correctness of PaintGeneralPolyhedron

Lemma IV:
The AC implies that if maxphi(E) > maxphi(V) with E and V e-nbrs, then E is a posterior e-nbr.
Proof:

This consequence of the AC is proved similarly to Case II of Lemma II. In Figure 4B, if phi(p1) > phi(p2) and E is prior to V, alpha must be greater than pi/2.▯

Lemma V:
The AC implies that a prior(posterior) e-nbr is prior(posterior) for all e-nbrs of equal maximum phi.
Proof:
This important consequence of the AC is proved similarly to Case I of Lemma II. In Figure 4A, if E were prior to V and posterior to P, beta would be less than pi/2.▯

Lemma VI:
No e-nbr E is PolyDangerous to a patch P.
Proof:
Case I: maxphi(E) /= maxphi(P):
by Lemma IV, the patch of larger maximum phi is a posterior e-nbr; it is correctly (by the comment on the Priority Test) painted first by PaintGeneralPolyhedron.
Case II. maxphi(E) = maxphi(P):
by Lemma V, no patch in any input set for PaintAPhi is both prior and posterior to its e-nbrs. Thus the prior and posterior patches may be classified by local examination and painted in correct relative order, as PaintAPhi does. Isolated patches overlap with other patches in the input set in at most a single value of theta, and can be painted independently.▯

Theorem VI:
PaintGeneralPolyhedron is correct; in well-tesselated polyhedra, no patch Q is PolyDangerous to a patch P.
Proof:
As for Theorem V, using Lemma VI instead of I.▯

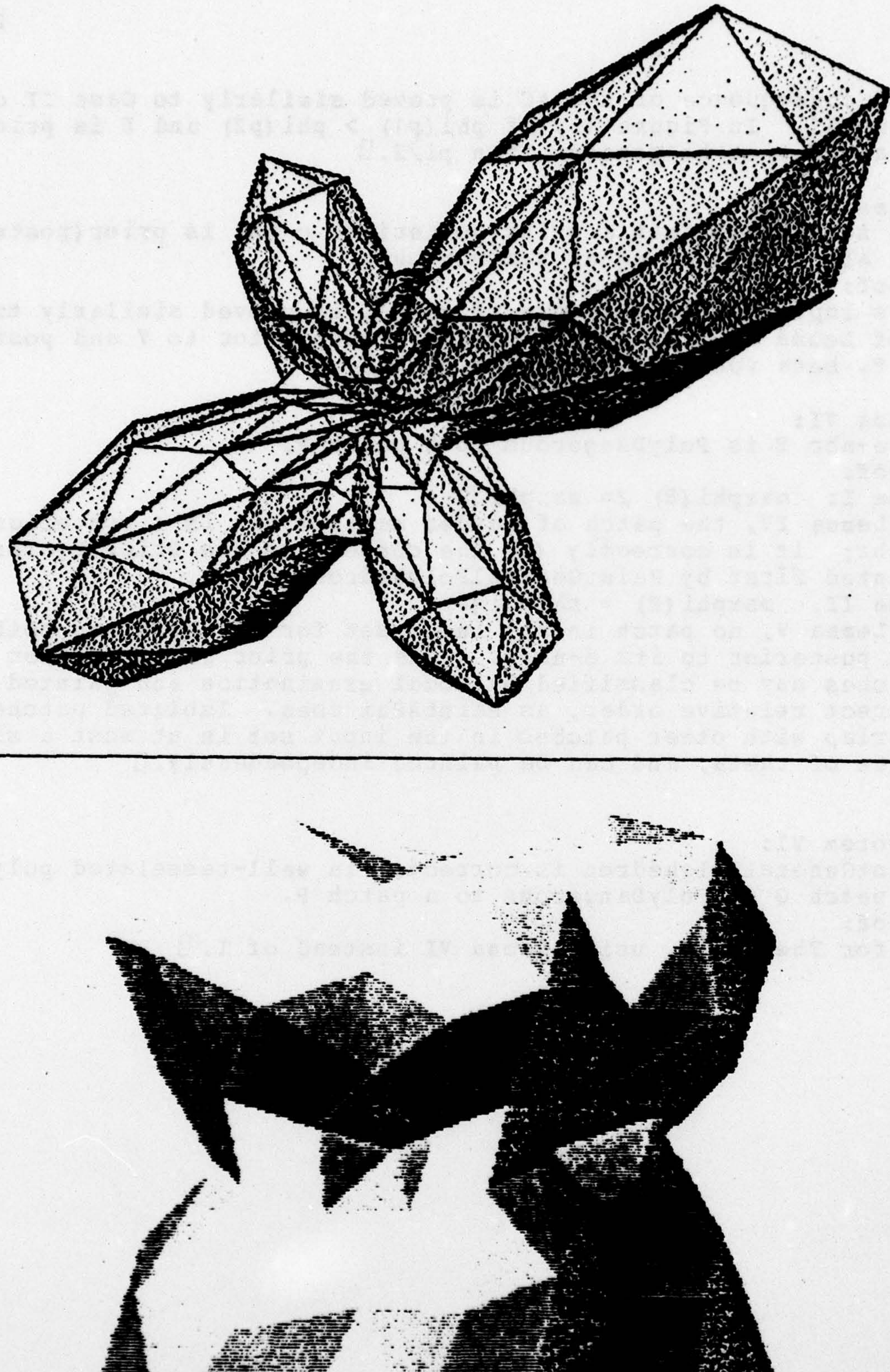
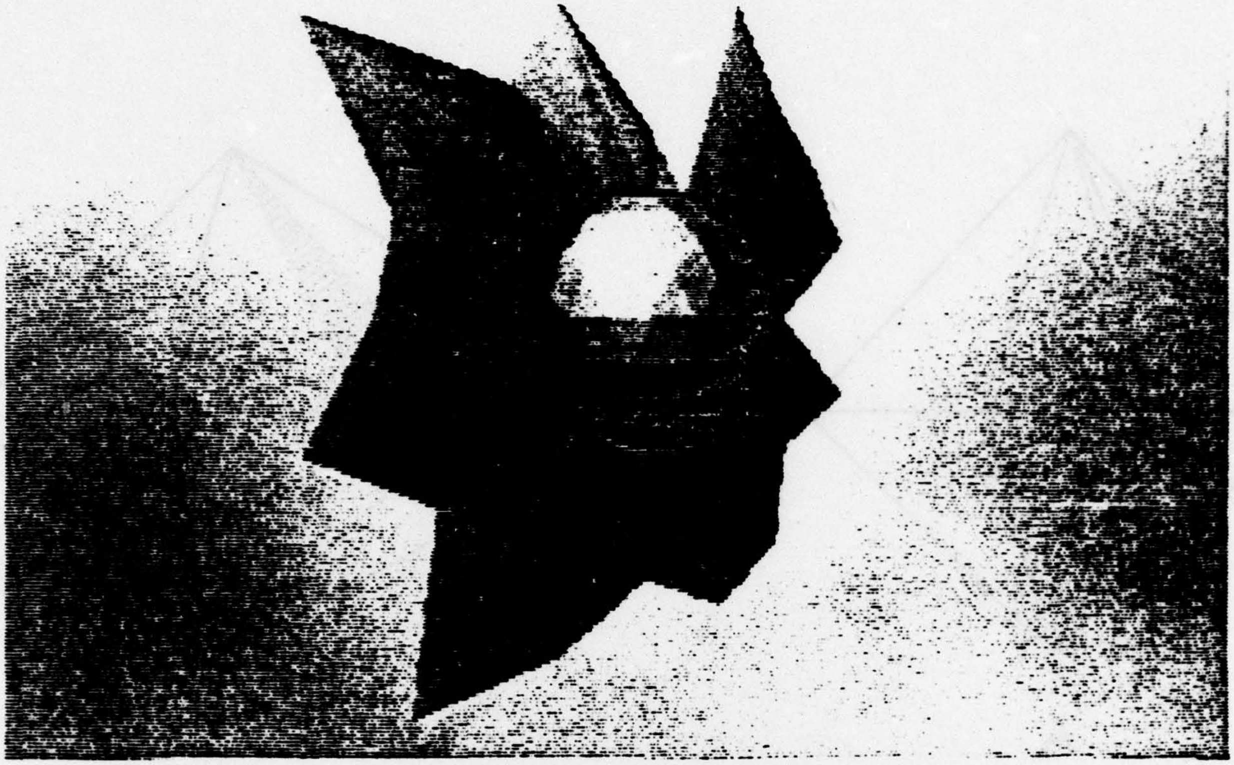


Figure 1



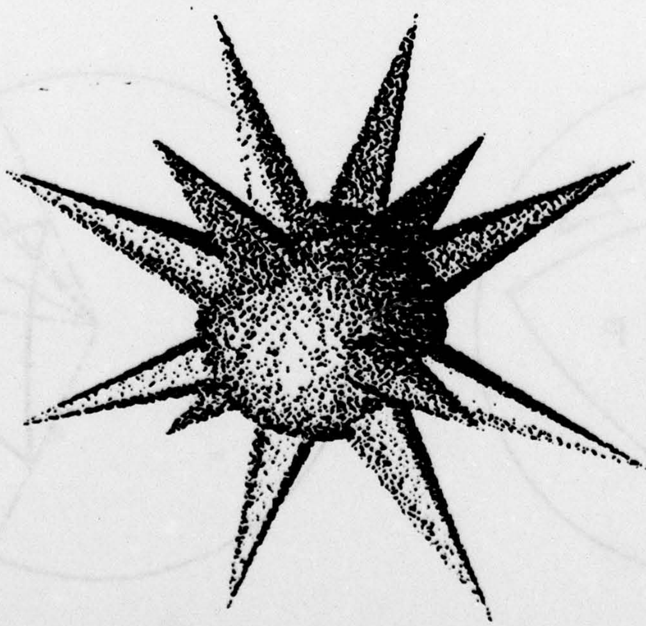


Figure 1 (Continued)

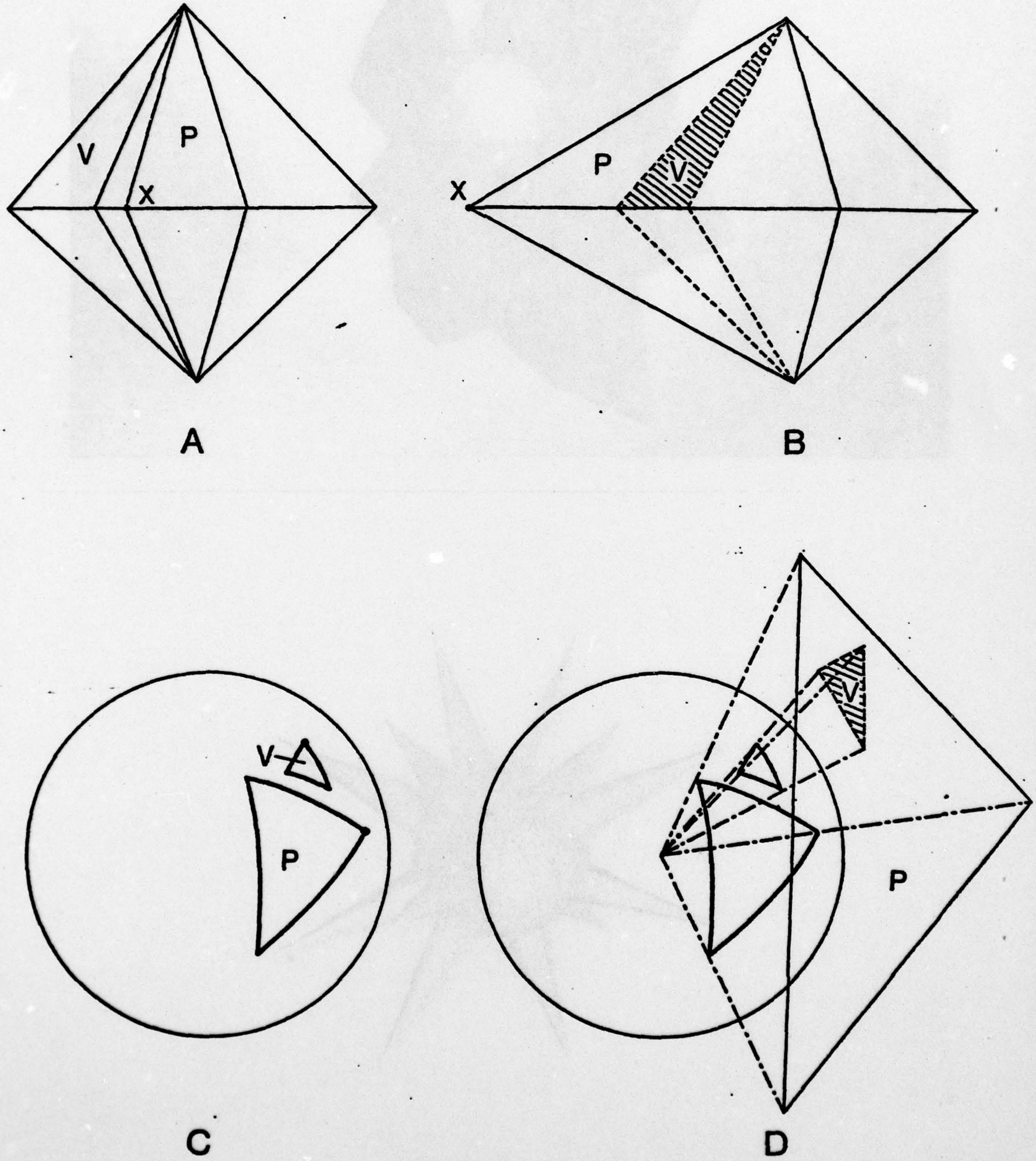


Figure 2.